プチコンmkII・プチコン3号を
一生遊ぶ本

ver. 1.1

プチコンmkII 永遠に！！！ プチコン3号

もくじ

はじめに

　プチコンmkIIは2012年3月14日にDSiウェアとして発売され（3DSでも使用可能）、プチコン3号は2014年11月19日に3DSアプリとして発売されており、**2023年3月28日のニンテンドーeショップがサービスを終了**によって新規購入ができなくなってしまいました。
　しかし、すでに購入してダウンロード済みのプチコンmkII、プチコン3号は一部機能に制限はあっても基本的に永続して使うことが可能です。しかし、快適に使うには準備が必要なのでそのことを本書でまとめてみました。

　私が作ったプログラムに関してはプログラムリストを掲載していますが他者のプログラムは著作権の関係上、公開キーのみの掲載となります。公開キーはただし、**2024年4月9日午前9時で任天堂の3DS、WiiUのオンラインサービスを終了予定となっています。それに伴いプチコン3号、プチコンBIGの公開キーも使用ができなくなるためできるだけ早くダウンロードすることをおすすめします。**

第1章　プチコンmkII編

1-1　プチコンmkIIで自作ソフトを公開する、ダウンロードをする

　プチコンmkIIはプチコンmkII内で生成されるPTCファイルをPCでQRコードに変換することで公開可能にしています。スマイルブーム公式のQRコード変換ツールはすでにサービスが終了していますが有志が作ったPTCUtilitiesはまだダウンロード可能なので今のうちにダウンロードしておきましょう。

PTCUtilities
http://micutil.com/ptcutilities/top.html
（※本書に記述のリンク先へのアクセスやソフトの使用は自己責任でお願いします）

　プチコンmkIIのプログラムはPTCファイルをパソコンでバックアップしておけば自作プログラムが失われることはありません。

　プチコンmkIIで作られたゲームやツールはQRコードで公開される（QRコード自体にプログラムが含まれている）ためQRコードさえ残っていればいくらでもダウンロードが可能です。興味があるプログラムにおいてはQRコードのバックアップをとっておくと容量が足りなくなって消してしまったプログラムを再度ダウンロードすることが可能なのでプチコ

ンmkIIは本体が壊れてプチコンmkIIそのものが使えなくなるまでプログラムは自由にダウンロードできます。まさに一生ものだと思います。(PTCファイルのバックアップでもいいけどQRコードをバックアップしておいた方がすぐにプチコンmkIIで使用可能になる)
　プログラムが入ったPTCファイルはパソコンのテキストエディタを使えば簡単に中身のプログラムを見ることができます。ただし、プチコンmkIIの独自の文字を使用している場合はこの方法だと正しく見ることができない場合があります。

　また、1画面プログラムならば編集画面の写真を1枚撮るだけで公開が可能です。興味がある人は後述の方法で1画面プログラムを作ってみてください。1画面プログラムならば入力もそれほど時間を掛けずに行えます。

1-2　プチコンmkIIのプログラムの作り方

　プチコンmkIIのマニュアルではあまり書かれていない使い方について書いていきます。

　プチコンmkIIで扱える数値は「(32bit)固定小数点方式」と呼ばれるものとなっています。これは一般的によく用いられている浮動小数点方式(プチコン3号では実数型と呼ばれている)と違って小数点の位置が固定となっていてそのため扱える数値の範囲はかなり狭くなっています。整数部が符号込みで20bit、小数部は12bitとなっていて、-524288から524287.999755(2の31乗-1を4096で割ったものであり、画面表示上は524287.999)までの数値で1/4096単位の数しか扱えません。1/4096単位未満の数は切り捨てられるため、1/4097を実行すれば0になるし、2/4097を実行すれば1/4096という値になります(ただし、画面表示は小数第3位までなので表示上は0となる)。そのため絶対値が小さい数ほど誤差が発生しやすくなります。**誤差を無くすには1/4096単位の数のみを使用する必要があります。**
　あと、100万とか、1億とかの大きな数を表現したければ複数の変数に分けて扱う必要があります。しかし、小数点の位置を変えれば32bit整数型として扱うことも可能になるため32bit固定小数点を32bit整数に変換するサブルーチンも用意しておきました。

固定小数点→32bit整数変換

```
@INT
Z=4096:IL=INT%(10000/Z)*Z:IH=0OR(INT-IL/Z)/10000*Z
INT$="-"*!IH*(INT<0)+STR$(ABS(IL)):INT$=STR$(IH)*!!IH+"0"*(4
-LEN(INT$))*!!IH+INT$
RETURN
```

32bit整数→固定小数点変換　(上記の逆変換)

```
@INT2
Z=4096:IS=LEFT$(INT$,1)=="-"IN=LEN(INT$)-IS:INT$=RIGHT$(INT
$,IN)IL=VAL(RIGHT$(INT$,4))/Z
INT=(1-IS*2)*(IL+VAL(LEFT$(INT$,(IN-4)*(IN>3))))/Z*10000
RETURN
```

使用例

```
A=12345/4096
B=A*56789
INT=B
GOSUB @INT
?INT$
```

12345×56789を行い701060205という結果を表示するプログラムです。

　サブルーチン@INTを使うには変数INTに表記したい値を予め1/4096にしておきます。1億を入れる場合はINTに24414.0625を入れておけばよいです。あとはGOSUB@INTとすることでINT$に″100000000″という文字列が入ります。
サブルーチン@INT2を使うには変数INT$に計算したい値を文字列で入れておきます。1億を入れるにはINT$に″100000000″を入れておけばよいです。あとはGOSUB@INT2とすることでINTに24414.0625という数値が入ります。
　これによって、プチコンmkIIで絶対値2147483647までの大きな数値を扱って自由に演算処理をすることが可能です。加減算を行う場合は予め4096で割ったものをINTに入れる必要がありますが、乗除算ではそれは不要です。
　簡単な足し算程度ができればいいのならば4桁ごとに異なる変数を用意すれば1億、1兆という大きな数値も疑似的に

扱うことは可能です。

　また、プチコンmkIIのセーブ機能はかなり特殊でプログラムやリソースはセーブできても変数の値を自由にセーブすることはできません。唯一セーブができるのは変数MEM$だけです。変数MEM$には256文字までの文字列を入れられるため例えばハイスコアベスト10を記録する場合にはデータ1つあたり25文字以内に抑えて桁そろえをして記録するかデータの区切り文字を入れて記録すると良いでしょう。
　256文字（256バイト）では全然足らないならば使っていないグラフィックページにドットを打っていき（1ドット当たり0～255の数値を記録可能）GRPリソースごとセーブすれば48KBまでのセーブデータに対応できます。しかし、ただでさえ少ないプチコンmkIIの保存領域を数KBのセーブデータのために48KBも浪費するのはもったいないので8KBで済むCHRリソースを使いたいところです。しかし、CHRリソースはGRPと違ってデータの読み書きを1バイトずつ行うのは非常に面倒です。一旦GRPに記録しておいてセーブの段階でCHRリソースに変換するのが最もお手軽なのでそれを実現するCHRセーバー、CHRローダーを用意しました。ともに1行プログラム（プチコンmkIIは1行に100文字まで入力可能）なので手軽に入力して利用できると思います。

CHRセーバー
```
FOR I=0TO 255FOR J=0TO 31Z=GSPOIT(I%8*32+J,I/8)C$=C$*!!J+HEX$(Z,2)NEXT:CHRSET R$,I,C$NEXT
```
CHRローダー
```
FOR I=0TO 255CHRREAD(R$,I),C$FOR J=0TO 31Z=VAL("&H"+MID$(C$,J*2,2))GPSET I%8*32+J,I/8,Z:NEXT:NEXT
```

　あとプチコンmkIIではプチコン3号とは異なり下画面を利用するためには専用命令が必要です。
　まずはPNLTYPE "OFF"で下画面のキーボードを消して自由に表示が行えるようにします。コンソール文字は下画面ではPRINT命令は使えないのでPNLSTR命令を使いますがこれは数値は表示できず文字列のみとなるので注意が必要です。グラフィック、スプライト、BGに関してはGPAGE 1、SPPAGE 1、BGPAGE 1のように引数を1とすることで下画面に表示することが可能になります（上画面に戻す場合は引数を0にする）。

　プチコンmkIIでINKEY$で文字入力を行う場合には文字が入力バッファに溜まっていて上手く判定ができない場合があります。そういう時は下記のようなバッファを空になる処理を実行すれば良いです。

```
FOR I=0TO""<INKEY$()I=0NEXT
```

プチコンmkIIではSELECTボタンは取得可能なのですがSELECTを押すとプログラムが停止します。しかし、プログラム起動前からSELECTを押している場合に限り下記のようなプログラムで判定すればSELECTボタンを判定できるのでデバッグモードにジャンプするなどの要素を入れることができます。

```
IF BUTTON()AND 2048THEN @DEBUG
```

　GRPにグラフィック表示を行う場合は60fpsを確保できず画面がちらつく場合があります。そういう時は使ってないGRPページに描画をし終わってからページ切り替えをするダブルバッファリングを行うことでちらつきを無くせます。
　ページ0とページ1を使用し、上画面に表示する場合は下記のようになります。

```
GPAGE 0,P,!P:P=!P
```

　プチコンmkIIはRボタン、STARTボタン、←ボタンを同時押しすることで画面クリアができるのですが、ホームメニューに戻ることでも画面クリアができます。
　ホームメニューから実行する場合は画面クリアだけではなく**すべての変数の初期化**も行います。したがって、「プログラムを作る」で実行する場合とホームメニューで実行する場合ではプログラムの挙動が変わる場合があるため注意が必要です。

1-3　プチコンmkIIの1画面プログラムの作り方

　プチコンmkIIの編集画面は**29文字×24行**となっています。**最大696文字記述が可能**なのですがプチコンmkIIのエディタは**1行29文字を超えると画面をスクロールさせないと全体が見れなくなります**。したがって、プチコンmkIIで1画面プログラムを作るには1行に記述する文字数を29文字以下に制限する必要があります。

文字数を減らすには単純に省略して減らす場合とプログラムを作り変えて減らす場合の大きく分けて2通りの方法があります。前者は機械的な単純作業で短縮できるためいくつかのパターンを覚えておくと良いでしょう。

プチコンmkIIではコロンやスペースが短縮できる場面が多いです。

例えばA=1:B=2はA=1B=2のようにコロンを省略できますがA=X:B=YはA=XB=Yのようにコロンを省略できません。これはXBが変数として認識されるためです。スペースもこのように別の変数や命令と認識されてしまわない場合に限り省略が可能です。

また、引数を省略できる場合も多いです。BEEPはBEEP 0を省略したものであり、VSYNCはVSYNC 1を省略したものとなっています。

あと論理否定「!」もプログラム短縮に貢献できる場面は多いです。

例えば!AはAが0の時に1、Aが0以外の時に0になります。!!AとすればAが0の時に0、Aが0以外の時に1になります。これを使えば、!(!!A*!B)は!A+!!Bのように短縮が可能です。(ド・モルガンの法則より)

プチコンmkIIは分岐やループ機能が貧弱です。ループはFOR～NEXTしか使用できません。GOTOを使った分岐はできるもののラベルは行の先頭に記述する必要がありラベルを記述した行にはラベル以外を記述できません。IFもELSEを使うことができません。したがって、条件判断を行いたい場合には29文字以内に収めるというのはかなり厳しく感じるかもしれません。

しかし、FOR～NEXTを上手く使うことでWHILE～WENDの代わりにも複数行記述が可能なブロックIFの代わりにもなります。

WHILE～WENDの代わり

```
FOR I=1TO (条件式)
(処理内容)
I=0
NEXT
```

ブロックIFの代わり（上記からI=0を省いただけです）

```
FOR I=1TO (条件式)
(処理内容)
NEXT
```

多重ループを使うことで複雑な条件分岐処理も可能になります。

プチコンmkIIでは1画面プログラムを作る場合には制限が大きいですが剰余の計算が%の1文字で可能なためこれを上手く活用すれば驚くほど短縮できる場合もあります。

剰余を使った短縮の例1　じゃんけんの勝敗

変数Xに自分の手（グー＝1、チョキ＝2、パー＝3）が入っていて変数Yに相手の手が入っている場合、IF X==1 AND Y==3 THEN～ などのようにすべてのパターンを記述しなくても(Y-X+3)%3の値だけで勝敗判定が可能です。(0はあいこ、1は自分の勝ち、2は自分の負け)

Bボタンがグー、Yボタンがチョキ、Xボタンがパー（←これは手の形状に似ているので覚えやすい）だとすると変数BにBUTTON関数の値が入っているならばX=96%B%5とすることでボタンを押すだけで自分の手を0、1、2のいずれかに変換可能です。

剰余を使った短縮の例2　十字ボタンを使った8方向移動

普通に作れば論理式を使って短縮化してもこの長さになります。

```
X=X-((B AND 4)==4)+((B AND 8)==8)
```
（※X方向のみ）

しかし、剰余を使えば同じものがここまで短縮ができます。

```
X=X-(B%16-B%4+1)%3+1
```

プチコンmkIIで1画面プログラムを作るには短縮によって合計の文字数制限をクリアするだけではなく1行29文字という制限もあるため順序を変えても問題ない部分を変えることでパズルのように各行を29文字に近づけるという操作が必要になることがあります。最初の段階でこんなことまで考える必要は無いですが、凝った1画面プログラムを作ろうとするとこれらは必要不可欠なテクニックになります。

では、短縮の効果がどれほどのものなのかはプチコンmkIIで作ったギタープログラムと概ね同じ動作を行う1画面版のギタープログラムを置いておきます。両者を比較すれば1画面プログラムの作り方が見えてくると思います。

PETIT GUITAR（1画面プログラム）

```
CLEAR:ACLS:GPAGE 1PNLTYPE"OFF
W$="[illegible]
X$="[illegible]
Y$="[illegible]
Z$="[illegible]
DIM C(999)Z=256A$=W$+X$+Y$+Z$
FOR I=0TO 47B$=MID$(A$,I*2,2)
X=ASC(RIGHT$(B$,1))*Z+ASC(B$)
X=X-257Y=X%16W=16FOR J=2TO 7
X=X-Y*(J<3)V=X%(W*4)/W:U=!!V
X=X-V*W:C(J+I*6)=V+Y*U:W=W*4
GFILL J*30,0,J*29,5+4,Z,7H=64
A(J)=J*5+118-(J>5)NEXT:L=I+40
BGMSET I+128,"T1@27N"+STR$(L)
NEXT:FOR X=0TO 9B=BUTTON()CLS
X=0OR TCHX/30R=Y>X?CHR$(C+H);
E=!!(128AND B)C=(112AND B)/16
S=X+R:D=0OR(C*5-4)/3+E?"#"*E
M=TCHY:IF 512AND B THEN P=C
O=(S>1)*(S<8)N=(M-H)*(M>H)?P
Q=C(!!C*(D*6+P*72)+S*O)*!!C
F=!!(X-Y)*T*U*O:G=Q+A(S*F)U=T
T=TCHST:IF F THEN BGMPLAY S,G
BGMVOL S*F,N:Y=X:WAIT 1NEXT
```

1/2　PRG：OCHAPGTR　2/2

プチコンギター

1/2　PRG：OCHAGTAR　2/2

　まずはシンプルなプログラムを作りそれを短縮することで1画面に収めるわけです。ギターコード（48種）のデータは独自の手法（1つの弦情報を3bitに圧縮、6本の弦情報を2文字に圧縮）で圧縮することで1画面に収まっています。
　もちろん、短縮だけでは1画面に収まらないことも多いため表示をさらにシンプルにするなど特に重要ではない部分を切り捨てること（取捨選択）も1画面プログラムを作る際には必要になることも多いです。

　私が作った1画面プログラムのゲームをいくつか掲載するので参考にしてみてください。1画面プログラムを作るには簡略化や様々な短縮テクニックも必要になりますが1行29文字の制限が結構大きいです。この制限のために処理の順序を入れ替えてできるだけ1行を29文字付近にするというパズル的なテクニックも要求されてきます。

　各ゲームのプレイ方法はQRコードを読み取ったプログラムの末尾に簡易マニュアルを記述しているのでそれを見てください。

プチコン100m走 mkII

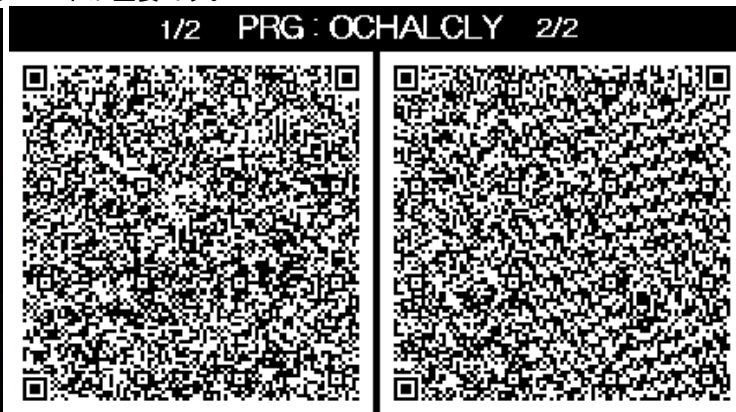
ABボタン連打で競う100m走ゲームです。目指せ世界記録。

```
W=999N=200ACLS:GPAGE 1GCLS 7
U=72GLINE U,0,U,U,15S$="SPU1
H=256R=109P=0GFILL 0,96,H,H,4
FOR I=0TO H*6J=I%6O=I%3*8V=50
Q=O-H+I%H*8GPSET RND(H),I%U,8
M=28912%(J+13)+!J*14G$="GOAL!
GPUTCHR Q+O,U+O,S$,U+I%4,0,1
GPUTCHR J*8,99,"BGF",M+65,4,1
NEXT:FOR E=0TO 1CLS:T=0A=32
SPSET 0,64,3,0,0,0SPSCALE 0,N
SPOFS 0,48,140SPANIM 0,4,4S=0
GPAGE 0GCLS 10?" "*H"WR "W/V
FOR X=0TO H*5B=BUTTON()X=X+S
IF!(A-B)THEN P=(4-S)/5A=48-A
S=P+S*0.97T=T+1Z=X%512/2Y=X%H
GCOPY 1,0,0,Y-1,55,H-Y,136,1
GCOPY 1,Y,0,255,55,0,136,1P=0
GCOPY 1,0,U,Z-1,R,H-Z,0,1
GCOPY 1,Z,U,255,R,0,0,1WAIT 1
FOR I=T TO 1?" "*75"READY ";
WAIT 99BGMPLAY 0E=0?"GO!
NEXT:X=X-1LOCATE 1,9?"[illegible] "T/V;
NEXT:W=W+(T-W)*(W>T)?" "*99G$
BGMPLAY 4WAIT 99LINPUT A$NEXT
```

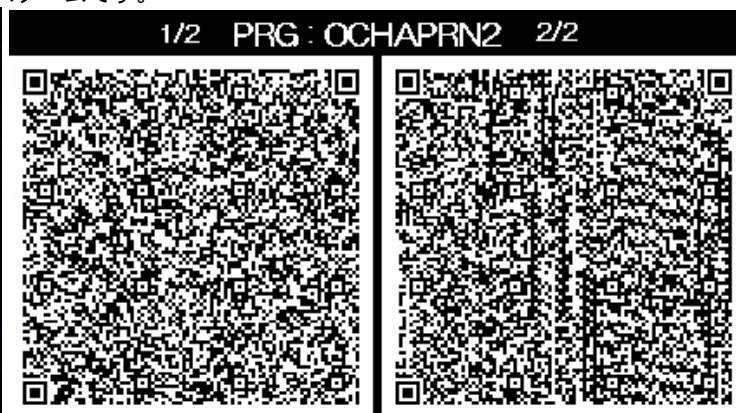
1/2　PRG:OCHAL100　2/2

プチコンクレー射撃

クレー射撃風のガンシューティングです。素早いリロードが重要です。

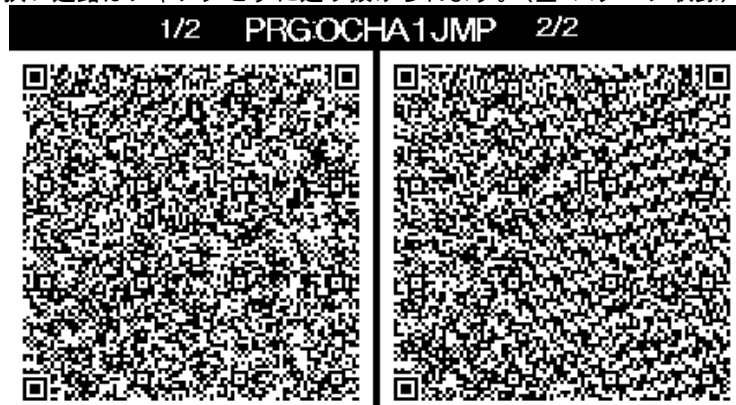
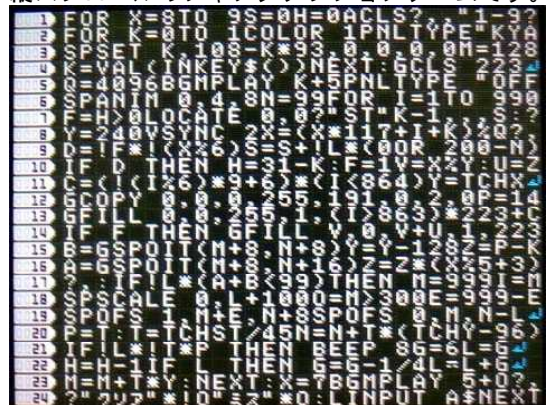
1/2　PRG：OCHALCLY　2/2

PETIT RUN mkII

シンプルだけどタイムアタックが熱い疑似3Dレースゲームです。

1/2　PRG：OCHAPRN2　2/2

JUMPING ISLAND

縦スクロールのジャンプアクションゲームです。狭い通路はジャンプせずに通り抜けられます。（全9ステージ収録）

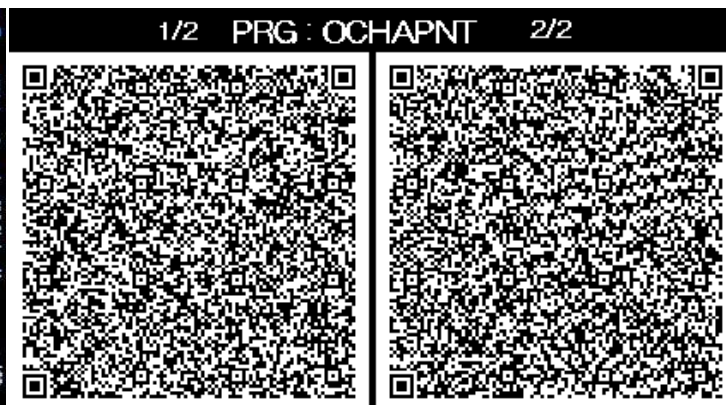
1/2　PRG:OCHA1JMP　2/2

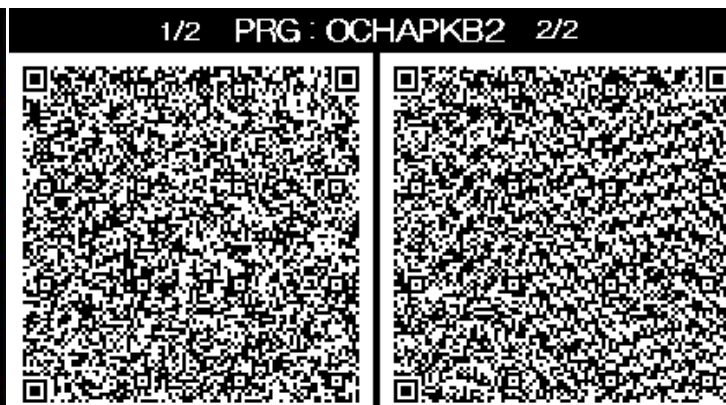
プチコン100m走mkII　プチコンクレー射撃　PETIT RUN mkII　JUMPING ISLAND

　プチコンmkIIの1画面プログラムの作り方について詳しくは既刊「プチコン 1画面プログラムのススメ」「プチコン 1画面プログラムのススメ ver.2.0」を御覧になってください。

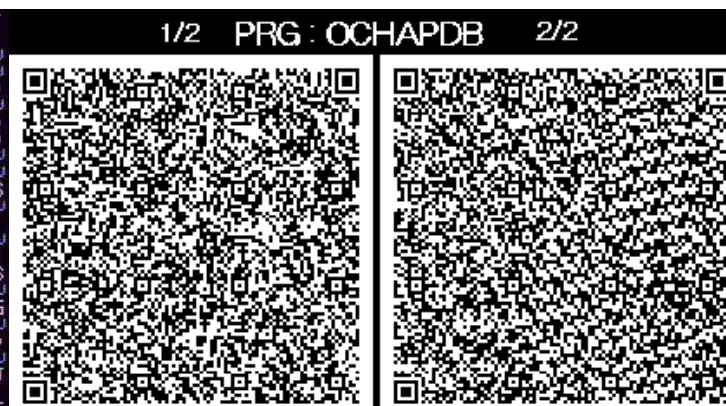
1-4　ぜひ入手しておくべきプチコンmkIIプログラム

　プチコンmkIIには数多くの大作、力作プログラムがありますが、著作権の観点から他人のプログラムはここでは掲載できません。タイトルだけを羅列しても意味が無いためここでは私が作ったプログラムをいくつか掲載しておきます。
　私の公開プログラムはすべてプログラムの末尾に使用方法やプレイ方法を記述しているので使用方法（プレイ方法）が分からない場合はQRコードを読み取ったプログラムの末尾にある簡易マニュアルを読んでください。

PETIT PAINT mkII（1画面プログラム）
シンプルで使いやすいお絵描きソフトです。

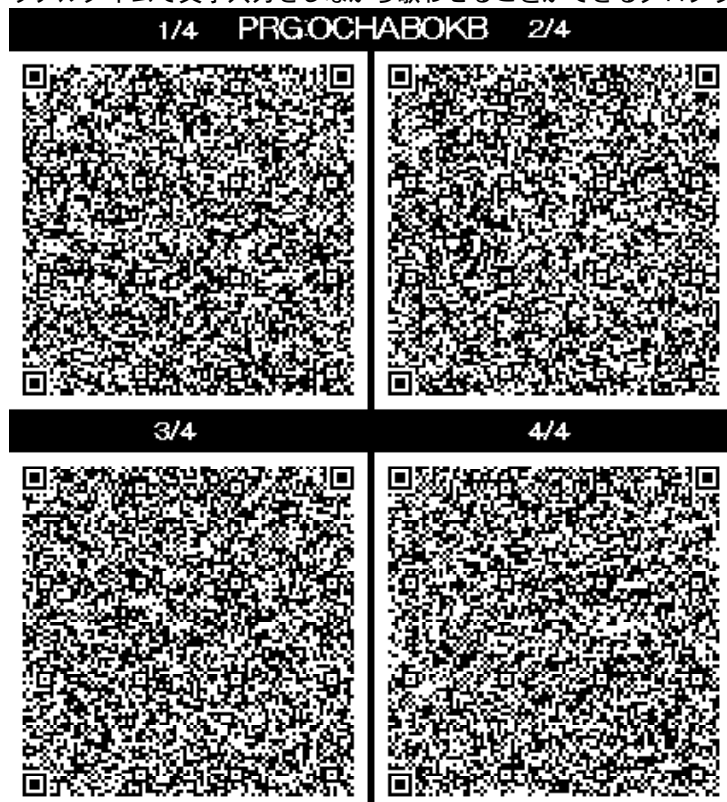
1/2　PRG：OCHAPNT　2/2

PETIT KEYBOARD mkII（1画面プログラム）
キーボード（鍵盤）演奏プログラムです。

1/2　PRG：OCHAPKB2　2/2

PETIT DRUM + BASS（1画面プログラム）
ドラムとベース演奏プログラムです。

1/2　PRG：OCHAPDB　2/2

前項のギターに加えてキーボード、ドラム、ベースがすべて1画面プログラムで揃ってしまうのでプチコンmkIIを持ち寄ればバンドを組むことが可能です。

棒歌ロイドキーボード
リアルタイムで文字入力をしながら歌わせることができるプログラムです。

1/4　PRG:OCHABOKB　2/4

3/4　4/4

棒歌ロイドキーボード　使用方法

発音する文字（文字はボタン操作で素早く入力可能）

バッファサイズを大きくすれば長い音を発音できる（反応とのトレードオフ）

鍵盤を押せばその高さで表示文字を発音する

文字入力方法についてはプログラムリスト末尾の簡易マニュアルをご覧になってください。

第2章　プチコン3号編

2-1　プチコン3号で自作ソフトを公開する、ダウンロードをする

プチコン3号で自作ソフトを公開する方法としてはスマイルブームのサーバーにアップロードをして公開キーを取得するのが一般的です。その公開キーを入力することでダウンロードが可能です。

しかし、2024年4月9日に任天堂による3DSおよびWiiUのオンラインサービスが終了予定となっています。そのため、冒頭にも書いたように4月9日以降はプチコン3号、プチコンBIGの公開キーによるファイルのやりとりができなくなります。したがって、現時点で公開されているプログラムはそれまでにダウンロードしておかないとダウンロードそのものができなくなるため早めにダウンロードしておきましょう。ゴールド会員権消失のため新規のサーバー枠を確保するため新規に作ったプログラム公開する場合にはすでに公開しているプログラムをサーバ上から消していたり、複数のゲーム、ツールを1つのプロジェクトフォルダにまとめたりして公開キーが変わっていたりしてすでに公開されている公開キーが無効になっている場合もあります。その場合もX（旧twitter）やpetitverseで探せば新しい公開キーが見つかる場合があります。あきらめずに頑張って探してみてください。

オンラインサービスが終了したらできなくなるのはダウンロードだけではなく当然アップロードもできなくなります。プチコン3号のプログラムを作ったのでみんなに楽しんでもらいたいという人は公開キーが使えるうちに早めに公開しましょう。公開しても4月9日以降はダウンロードできなくなるのですが、それはやむを得ないでしょう。

もちろん、オンラインサービスが終了したからといってプチコン3号そのものが使えなくなるわけではありません。オンラインに関係する公開キーを使ったファイルのやり取りができなくなるだけです。したがって、プログラムを作ったり、プログラムリストが公開されている他人のプログラムを手入力すれば他人のプログラムであっても問題なく使うことが可能です。

ちなみに、前述のようにプチコンmkIIならばQRコードを使うことでスマイルブームとは関係なく自作プログラムの公開やダウンロードが可能です。公開キーが使えなくなった後でのプチコン3号のプログラムの公開の仕方ですが、基本的にプログラムリストをすべて公開するという形であれば何の問題もなく公開が可能です（自作のキャラ、グラフィックを含む場合は難しいけど）。プログラムリストの公開にはプチコン3号のプログラムをパソコンに転送したり、プチコン3号上でQRコードに変換してそのQRコードをスマホやPCで読み取るなどの方法があります。いずれの方法も公開のた

めのハードルが少し高くなってしまうため個人的におすすめしたいのが1画面プログラムです。

プチコンmkIIとは異なりプチコン3号は標準でスクリーンショット撮影機能（SELECTボタン長押し）があるため1画面プログラムならば編集画面のスクリーンショットを1枚撮るだけで公開が可能です。興味がある人は後述の方法で1画面プログラムを作ってみてください。（プチコン3号の最新バージョンはver.3.6.3なのでスクリーンショットの撮影ができないという人は最新版がダウンロードできるうちにバージョンアップをしてみてください）

しかし、ある程度の規模のプログラムや自作のグラフィックを使ったプログラムの場合は編集画面のスクリーンショットでの公開では困難です。そこで多少ハードルは高くなってしまうもののパソコンに転送して公開する意外に方法はないです。
プチコン3号からパソコンへプログラムを転送する場合には画像経由（スクリーンショット等）、音波経由などが主な選択肢となります。プチコン3号では1ドットあたり15bitの情報を記録できるのですが、スクリーンショットを保存するとなるとJPEG化によって情報が失われるため1ドットあたり4～5bit程度に抑えておくと良いでしょう。この場合はパソコン側で画像から必要な情報の変換するソフトを別途用意する必要があります。
規模が数KBと比較的小さなプログラムならばQRコード経由の転送も可能です。QRコード化するプログラムは2-5で紹介しています。なお、本体を複数台所持している人はローカル通信ができるプログラムをそれぞれの本体に入れておくと転送が楽になります。
プチコン3号からパソコンへの転送は「QRコード経由」「画像経由」などいくつかの選択肢（使用するプログラムは2-5の通信の項目を参照）があるのですがパソコンからプチコン3号の転送は実質音波経由しかありません。90年代にインターネットに接続する際は電話線にて音波を使ってデータ転送していたし、80年代は自作プログラムの保存はカセットテープを使うことがあったのですがこれも音波を使った転送です。

プチコン3号からパソコンに転送の場合は3DSはイヤホン、パソコンはマイクに接続し、パソコンからプチコン3号に転送の場合はパソコンはイヤホン、3DSはマイクに接続すればOKです。しかし、この場合はプチコン3号とパソコンのケーブル（3DSに接続側は4極のケーブル）に接続する必要（ケーブル無しでも超低速ならば転送が可能）な上に転送ソフトも別途必要です。

いくつか開発されていますが最も有名なのはれい氏のPetitModemです。

http://rei.to/petitmodem.html

詳しい使い方についてはリンク先を読んでみてください。ちなみに1200bpsだと1秒当たり1200bit（150バイト）なので1MBのプログラムやデータを転送するには単純計算で約2時間かかります。

2-2　プチコン3号のプログラムの作り方

プチコン3号のマニュアルではあまり書かれていない使い方を中心に書いていきます。

プチコン3号はノーマル3DS（2DS）だけではなくNew 3DS（New 2DS）でも動作が可能ですがNew 3DSの方がノーマル3DSより3～4倍高速に動作します。そのためNew 3DSで開発してギリギリ60fpsのゲームだとノーマル3DSでは20fps以下の速度になってしまいます。そのためプチコン3号では動作している本体がNewかノーマルかを見分けるＨＡＲＤＷＡＲＥ関数が用意されています。ノーマル3DSの場合は自動的に軽量バージョンが立ち上がるようにするとかも可能です。

プチコン3号では数値は整数型と実数型を選択できます。整数型は「32bit符号付きの整数」なので-2147483648から2147483647までの範囲の整数を扱うことができます。実数型は「64bitの倍精度浮動小数点数」（C言語だとdoubleに相当）なので-1.797693134862315７e+308から1.797693134862315７e+308の範囲の数（有効桁数は15桁）を扱うことができます。標準では実数型で接尾辞として％を付けることで整数型を使えるようになります。また、ＯＰＴＩＯＮ ＤＥＦＩＮＴを実行することで整数型を標準にすることが可能です。この場合は実数型を使うには接尾辞として＃を付ければよいです。なお、Ａ＃とＡ％は異なる変数扱いになるので注意が必要です。
実数型で小数を表現した場合には2の累乗分の1を単位に表せる数値以外は誤差が発生します。正確な値を知りたい場合は私の自作関数のＰＳＴＲ＄関数（後述）を使用すると良いです。実数型変数の数値を文字列に変換する場合において標準のＳＴＲ＄関数では100万未満の整数値以外は誤差が発生します。そのためセーブデータの生成等の誤差ゼロでの文字列化が必要な場面でＰＳＴＲ＄関数が有用になってきます。
文字列変数は内部では配列変数として扱われておりＺ＄＝"ＡＢＣ"の時にＺ＄［０］は"Ａ"、Ｚ＄［１］は"Ｂ"、Ｚ＄［２］は"Ｃ"となっています。また、配列変数においては参照型が用いられています。Ｘ＄＝"ＡＢＣ"の時にＹ＄＝Ｘ＄とするとＸ＄もＹ＄も同じ"ＡＢＣ"という文字列を参照するようになります。そのため、Ｘ＄［０］＝"Ｄ"とし

てX$の値を"DBC"に変更するとY$もそれを参照して"DBC"という値になります。再びX$="ABC"のようにX$に新しい値を代入すれば参照を切ることが可能です。

プチコン3号ではプログラムスロットを複数使ったプログラムを作れます。その際に役立つのがUSE命令やVAR関数です。VARは主に変数の宣言に使用されますがVAR関数として使うことでVAR("A")のようにすれば変数Aの値が分かります。これを使えば、VAR(A$)とすることでA$に入っている変数名の値も分かります。同じ変数名でもプログラムスロットが異なれば別の変数扱いとなります。動作しているスロットと異なるスロットの変数はスロット番号に変数名を付けると値が分かります。プログラムスロット1の変数Aの値はVAR("1:A")で分かります。ただし、この処理の実行前に指定するプログラムスロットのコンパイルを行っておく必要があります。EXEC命令を使えばコンパイルをした後にそのプログラムスロットを実行するのですが、コンパイルのみを行いたい場合はUSE命令が有用となります。RUNをすれば全てのスロットの変数が初期化されますが、EXECやUSEの場合は指定したスロットの変数のみ初期化されます。
ちなみに現在動作中のスロットを取得するには私の自作関数のSLOT関数（後述）を使用すると良いです。

プチコン3号ではmkIIでは使えなかった自作関数が使えます。
自作関数において注意すべき点はローカル変数とグローバル変数の使い分けです。自作関数内で宣言している変数はすべてローカル変数になるのですがグローバル変数を自作関数内で使用する場合にはその自作関数よりも前にグローバル変数の宣言もしくは変数の記述を行う必要があります。引数に使用しているローカル変数の型は引数の変数に値を導入した時点で型が決まります。接尾辞に関係なく引数に整数型の値を入れれば整数型として処理され、実数型の値を入れると実数型として処理されます。
しかし、プチコン3号には型を判別する関数は用意されていません。そのため型判別を行いたい場合は私の自作関数のSUFFIX関数（後述）を使用すると良いです。
ローカル変数は代入した時点で型が決まるという特性を使い自作関数であるSUFFIX関数を組み合わせることで型の指定が不要なINIT関数を作ることができます。例えば要素数1の変数A[0]を定義すると、A=INIT(123)とすればA[0]には123という整数型の値が入り、A=INIT(123.0)とすればA[0]には123という実数型の値が入り、A=INIT("ABC")とすればA[0]には"ABC"という文字列が入ります。

INIT()関数　配列変数を型指定不要で使用できる

```
DEF INIT(A)
 VAR V$[0],V%[0],V#[0],Z[0]
 Z=VAR("V"+"$%#"[SUFFIX(A[0])])
 PUSH Z,A
 RETURN Z
END
```

自作関数内からジャンプする場合はすべてその自作関数内にあるローカルラベルへのジャンプとなります。自作関数外にあるグローバルラベルへのジャンプはラベルの前にスロット番号を指定してください。現在動作しているのがスロット1ならば自作関数内から自作関数外にあるプログラムスロット1のラベル@ABCにジャンプするにはGOSUB"1:@ABC"でジャンプできます。GOSUBは異なるプログラムスロットにジャンプした場合でもRETURNで自作関数内に戻れますが、GOTOでジャンプした場合には自作関数内に戻ることができずエラー発生の原因になるため自作関数内からGOTOでのジャンプは厳禁です。

さらに詳しく知りたい場合は既刊の「プチコン3号 裏マニュアル」「プチコン3号 自作関数完全マニュアル」を御覧になってください。

2-3　プチコン3号の1画面プログラムの作り方

プチコン3号の編集画面は46文字×29行となっています。したがって、これに収まるプログラムが1画面プログラムで**最大1334文字**記述できます。プチコン3号はプチコンmkIIとは異なり1行に書ける文字数に制限はほぼありません。
また、WIDTH 16とすれば文字サイズは縦横2倍（4倍角）になります。編集画面でも文字サイズは4倍角のままなのでこの状態で1画面に収まるプログラムも1画面プログラムとなります。この場合は**21文字×14行で最大294文字**となります。概ね4分の1に文字数が減っているのでこのWIDTH 16使用時の1画面プログラムはQSP（Quarter Screen Programの略）と呼ばれています。（「QSP」はプチコン3号界隈におけるユーザー同士でのみ通用する名称）

プチコン3号はプチコンmkIIとは異なりループに使用できる命令も多くラベルも自由な場所に記述ができるため1画面プログラムを作るためのハードルはプチコンmkIIよりも低いです。したがって、通常の1画面プログラムでは初心者で

も簡単に作れてしまうためあえて文字数の制限が大きいQSPに挑戦してみても良いでしょう。
　とはいえ、QSPは単に文字数制限があるだけでそれ以外は普通に作る場合と何ら変わりません。ただし、何も考えずに作ってしまうと簡単に文字数制限を超えてしまうし、その中で完成度を求めると知っておくことがたくさん出てきます。

1画面プログラムを作る流れ

1　まずはシンプルなゲームやツールのアイデアを考える
2　そのアイデアを要素ごとに羅列して3～4段階の優先度の設定をする
3　コーディングする
4　制限文字数をオーバーしたらプログラムリストの短縮を行う
5　それでもオーバーするならば優先度が低いものから切り捨てる
6　プログラムリストの短縮をする
7　文字数に余裕ができたら切り捨てたものの中から優先度が高いものを復活させる
8　4～7のループを行いこれ以上は無理と思ったら完成

　ちなみに優先度は3段階ならば**Aランク**「必要不可欠な要素（絶対に実装）」、**Bランク**「重要な要素（可能な限り実装）」、**Cランク**「あった方が良い要素（余裕があれば実装）」という感じで分類するのがベターです。Aランクは無ければそのゲームが成立しないので絶対に実装することが求められるため短縮してもAランクの要素がすべて実装できない場合はネタをさらにシンプルにするか未来の自分に託すなどをしてあきらめることも肝心です。作り慣れたり革新的な短縮のアイデアが浮かべば実現が可能になることもあるのであきらめたネタはその時が来るまで温めておくと良いでしょう。(カラー対応のお絵かきソフト「PETIT PAINT 2 QSP」は抜本的な短縮のアイデアを思いつくのに2年かかりました)

　では、実際にQSPを作るまでの流れをみていきましょう。
　例えばこれは周回型の見下ろしタイプのレースゲームのアイデアですが、見てのようにAランクの要素が多くて普通に作ったらとてもQSPには収まりません。

QSPのレースゲームの優先度設定

Aランク　コースの表示
Aランク　タイムの表示
Aランク　スライドパッドの左右で車を回転
Aランク　アクセルとブレーキの処理
Aランク　コース外の判定
Aランク　コース外に車が出た時の減速処理
Aランク　周回判定
Aランク　3周したらゴールになる判定

Bランク　周回数表示
Bランク　コースは攻略性の高いレイアウトにする
Bランク　車の加減速処理は攻略性の高いものにする
Bランク　リトライ機能
Bランク　速度によって変わるエンジン音
Bランク　ゴール時は音か画面表示で知らせる

Cランク　コースの見た目に拘る
Cランク　タイム表示の成形

　今回のプログラムでは未実装になった要素
Cランク　速度表示（未実装）
Cランク　スタートラインを表示する（未実装）
Cランク　スタート時のシグナル（未実装）

※ランクは私の中での設定であり制作する人が自由に設定して良い

　レースゲームを作った経験がある人ならば分かると思いますが、**車の挙動とコースの周回判定というのは結構長くなるので、普通に作ったらそれだけでQSPからあふれるくらいの文字数**になってしまいます。そのためQSPの範囲内でレースゲームとして成立させる（Aランクの要素をすべて実装）だけでも抜本的な短縮が必要になってきます。

前述のように自分で決めたAランクとなる要素はすべて実装しないとゲームとして成立しなくなるため必ず実装しなくてはなりません。**Bランク**も可能な限り実装しましょう。**Cランク**は実装してもしなくてもゲームには支障がないものですが、完成度を高めるためにはできるだけ実装しておきましょう。つまり、要素の取捨選択が重要なのです。

とはいえ、先ほども書いたように今回の場合はQSPの範囲内でAランクの要素の実装するだけでもかなり難しいです。ちなみに下記が完成したSIMPLE RACING QSPです。なんとAランク、Bランクのすべての要素に加えてCランクの一部の要素も実装しています。

使用方法：Aボタンを押せばアクセル、離せばブレーキ。3周走ってタイムを競え！（※公式レコード15.22秒）

これがそのSIMPLE RACING QSPのプログラムを分かりやすく記述したものです。

```
WHILE 1
 ACLS:BACKCOLOR RGB(30,93,0)
 '初期設定
 K=0 '周回数
 T=0 'タイム
 Z=0 '周回判定変数

 '自車の初期設定
 X=255 'X座標
 Y=30  'Y座標
 A=0   '向き(※画面上で右方向の時は0)
 S=0   '速さ

 '自車スプライト設定
 SPSET 0,352
 SPHOME 0,8,8

 'コース表示
 GFILL 9,9,390,230,RGB(96,104,128)
 GFILL 40,50,330,160,RGB(0,0,0,0)

 WHILE K<4  '周回数が4未満ならばメインループを繰り返す)
  LOCATE 0,0:?K '周回数表示
  LOCATE 4,0:?T/50
   'タイム表示(※本来はT/60にすべきだけどQSPに収めるためT/50としている)
   '(※T/50とすることで小数第2位までで割り切れるためタイム表示の整形処理が不要になる)

  T=T+1 'タイム加算

  '周回判定部分
  IF Y<128 THEN
   IF Z<256 && X>256 THEN K=K+1
    'スタートラインをY<128の範囲で左から右に通過の時は周回数に1プラス
   IF Z>256 && X<256 THEN K=K-1
    'スタートラインをY<128の範囲で右から左に通過の時は逆走とみなし周回数に1マイナス
  ENDIF

  STICK OUT V,W
```

```
  A=A+V/25 '自車の向きを計算(※単位はラジアン)
  B=BUTTON()
  G=GSPOIT(X,Y) 'Gの値はコースの状態(※0の時はコースアウト)

  IF B AND #B || B AND #X || B AND #Y THEN
   IF G!=0 THEN
    S=(S+(1/13))*0.98
     'アクセルボタンを押したときの通常の加速処理
     '(※Sの値は上限3.7程度で収束するため上限の判定は不要)
   ELSE
    S=(S+(1/13))*(0.98-1/7)
     'アクセルボタンを押したときの加速処理(コースアウト時)
     '(※Sの値は0.4程度で収束するため上限の判定は不要)
   ENDIF
  ELSE
   IF G!=0 THEN
    S=S*0.98
     '通常の減速処理(※Sの値は0で収束するため下限判定は不要)
   ELSE
    S=S*(0.98-1/7)
     'コースアウト時の減速処理(※Sの値は0で収束するため下限判定は不要)
   ENDIF
  ENDIF

  Z=X '周回判定用に1フレーム前のX座標を別の変数に保存

  '自車の移動
  X=X+COS(A)*S
  Y=Y+SIN(A)*S

  'エンジン音(速度によって高さを変えることで耳でも速度を確認可能になる)
  BEEP,S*512-2000

  '自車表示
  SPOFS 0,X,Y

  '画面上の自車の向きを変える(?SPROTは「度」なのでDEGで「ラジアン」から変換)
  SPROT 0,DEG(A)
  VSYNC
 WEND

 'ゴール処理
 BEEP 42
 LOCATE 0,1:?"@GOAL"

 'Aボタン入力待ち
 REPEAT
 UNTIL BUTTON()AND #A

WEND
```

全く同じ動作をするプログラムでコメント付きなので両者を見比べたらどのように短縮されているかがよく分かると思います。とはいえ、短縮前のこのプログラムも周回判定などの部分で処理の簡略化がすでに行われています。

そのため周回判定だけはコメントだけでは分かりにくいので解説をしますが、これはコースを3分割していると考えれば分かりやすいです。コースの下側（Y≧128の部分）をエリアB、コース上側（Y<128の部分）でスタート時より右側（X>256の部分）をエリアA、スタート時より左側をエリアCと考えるとA→B→C→Aの順番に通過した場合に1周したと見なしています。このように3つのエリアに分けることで周回カウントの漏れもなくなり逆走判定も可能になります（逆走してスタートラインを横切れば周回数はマイナス1となる)。

周回判定の方法
エリアA→エリアB→エリアC→エリアA
という順番で通過した際に1周したと見なされる

「Cの次にAを通過していればOK」「Aの次にCを通過していれば逆走と見なす」とすることでA→B→C→B→C→Aという順番であっても周回のチェック漏れが起きず、コースのいかなる場所を通っても確実な周回判定ができるようになる。

ちなみにコースの幅が4つの辺で異なっているのは速度コントロールやライン取りの重要性を出すためでこうすることで長方形のみで構成というシンプルなコースながら攻略性が高まっています。(下辺の幅広い直線は最高速を出せるためのストレート、左辺の狭く短い直線は急カーブっぽい扱いとなっていてここへの進入速度やライン取りがタイムにかなり影響を与えている)

アクセルとブレーキはAボタンを押すか離すかで判定することでシンプル化していますが速度の増加量が一次式ではないため増加時の上限判定や減速時のゼロ判定が不要になりプログラムの短縮につながっているだけではなくスタート時の加速が良く上限に近づくことで徐々に増加量が減るためスピード感があり気持ちの良い操作が可能になります。

リトライ処理は1回のゲームのプレイ時間が短いゲームならばQSPであっても実装して欲しい要素なのですが、実装だけならば非常に簡単でLINPUTを使用するだけです。LINPUTを実行すると文字入力待ち状態になりますがLINPUTはINPUTとは異なり空入力でもエラーにならないためこれによりダミー変数A$を用意することでAボタン入力待ちを簡単に実装できるのです。
さらにこのゲームではLINPUTのガイド文字列を使ってGOALの文字を画面に表示しているのですが、プチコン3号では@を付ければダブルクオーテーションで挟まなくても文字列扱いになるためLINPUT@GOAL;A$で「@GOAL」を表示してAボタン入力待ちになり1文字の短縮を実現しています。ギリギリの文字数で作っている場合は1文字の短縮でも馬鹿にできないのです。
なお、リトライはプログラムの最初から実行する必要があるのですが。GOTOでジャンプさせるとラベルが別途必要になります。そのためこのプログラムではEXEC.（EXEC 0の短縮形）で実現しています。こうするとスロット0でしか実行できないプログラムになってしまうのですが、「プログラムを見る」から実行すると自動的にスロット0になるため不都合はないでしょう。

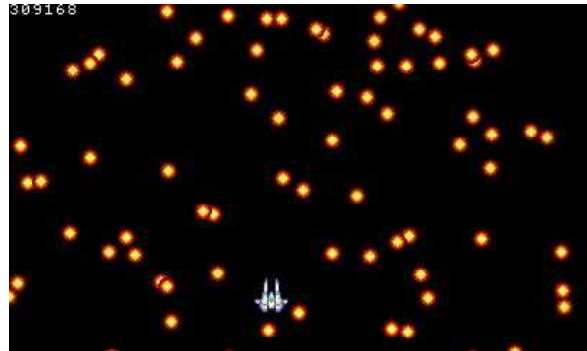
このようにかなり強引な処理を行うことでQSPに限界までの要素を盛り込んだ力作となっています。QSPにどこまで要求するかは人それぞれですが、限界まで作り込んでみたいという人は参考にしてみてください。
コース外の芝生をイメージしたBACKCOLOR -3E8はこのQSPを公開した当初は入れることができず、プログラムリストの短縮によって何とか入れることができました。当初は芝生の部分が黒だったことを考えると緑色になるだけでかなりイメージが変わると思います（Cランクの「コースの見た目に拘る」の実装)。この緑色を表示するのにRGB(30,93,0)とすれば文字数は増加して入れることができなくなるため浮動小数点表記を用いて短縮が行われています。

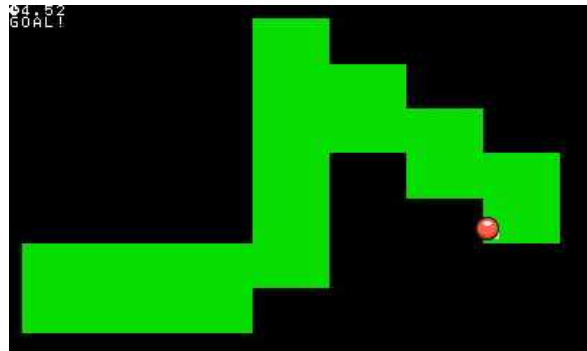
1画面プログラム（QSPを含む）を作る際には2つのアプローチがあります。まずは、今回のように必要最小限の要素（Aランクの要素）でさえも文字数制限で入れることが難しいので最初の時点でプログラムの短縮が要求されるタイプです。そして、もう1つは必要最小限の要素（Aランクの要素）を全部入れたとしても特に短縮せずとも1画面に収まるためそこから肉付けしていくタイプの2つです。前者と後者では「Aランクの要素の量が異なる」という違いしかないのですが、1画面プログラムに作り慣れていない人であれば後者の方が作りやすいと思います。というのもAランクの要素をすべて入れればその時点でゲームとして成立するので「完成度の高さ」を抜きにすれば「1画面プログラムを完成させる」という面ではハードルが低くなるためです。
1画面プログラムを作り慣れていない人はAランクの要素を全部入れても1画面に収まるようなすごくシンプルなアイデアをまずは考えましょう。そのためには「シンプルすぎるから駄目」ということはなく「シンプルだから良い」という考えを持つことが必要になります。1画面プログラムはプログラムが肥大になりがちな人にもオススメできます。

1画面プログラム（QSP）の作り方をもっと詳しく知りたい場合は既刊の「プチコン3号 QSP完全マニュアル」「プチコン3号 QSP完全マニュアル PART2」「プチコン3号 QSP究極完全マニュアル」を御覧になってください。

以下、私が作ったQSPのプログラムをいくつか掲載しておきます。これらのQSPは1画面分のプログラムを入力すれば実行可能であるため本書が手元にある限りは本体からプログラムを消しても(再び入力するだけで)何度でも入手が可能になります。ごく当たり前のことですが、それが簡単にできるのがQSPの良いところです。

QSPゲーム

DANMAKU QSP

弾幕避けゲー。スライドパッドで移動してひたすら避けまくって高得点を目指せ。

玉転がしゲームQSP

本体を傾けてコースから落ちないようにボールをゴール（白の四角）まで転がしてタイムを競え。

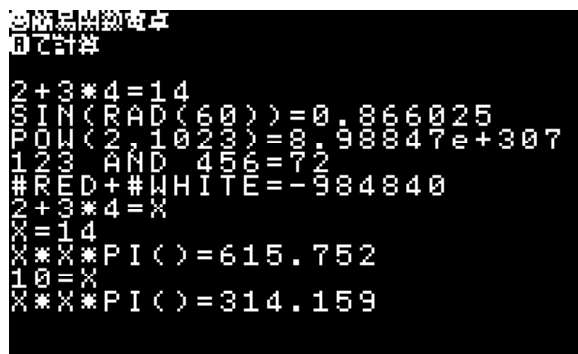
QSPツール

PETIT PAINT 2 QSP

カラー対応のお絵描きツール。十字ボタンで太さ調整（押さない→上→下→左→右の順で太くなる)、色変更はX、スポイトはLもしくはRを押しながらタッチ、セーブはR+YもしくはL+R、ロードは起動時に自動的に行う。ファイル名は"_" なので上書きされたくない場合はファイル名を変更する。

パレットGRPメーカー2

PETIT PATINT 2 QSP用のパレットを生成するツール。起動したらRGBの階調数を入力すればパレットを生成するのでセーブしてPETIT PATINT 2 QSPで利用可能になる。
色数は各階調数を3乗した色数となる。4を入力したならば64色。5ならば125色、6ならば216色のパレットを生成する。パレットと描画面は同一レイヤー（全体で1レイヤー）なので注意する。

簡易関数電卓QSP

式を入力することでその値を計算する。プチコン3号のすべての演算子、関数が使用可能。F1に答えが入り、F2に直前に入力した計算式が入る（Bボタンを押しながら計算するとF3に計算式が入る）、Xを押しながら式や値を入力するとその値が変数Xに入る。自作関数も使用可能なので自前で様々なプログラムを用意すれば自分好みの関数電卓、OSもどきのような使用も可能になる。（詳しい使い方の説明やオプションライブラリはダウンロード版の簡易関数電卓QSPに含まれている）

連射計測QSP

連射速度を計測するツール。10秒間ひたすらAボタンを連打。めざせ秒間16連射！

あいのてQSP

音声入力によって話し相手になってくれる人工無能。（といっても相槌を打つだけ）

他にも私はプチコン3号で多数のQSPを作っています。下記の公開キーでダウンロードされるプロジェクトフォルダ内にある.MENUを実行すればメニュープログラムが実行されます。（このメニュープログラム自体もQSP）
なお、使い方はダウンロードした各プログラムの末尾に記述しています。

おちゃめQSPコレクション　公開キー【 C3X3EJGF 】

2-4　便利な自作関数

ここでは、私が作った多数の自作関数の中からプログラムを作る際に有効活用できそうなものを厳選して掲載しておきます。(「これは俺が作った、俺が著作権を持つ自作関数だ」とか言わない限り）自由に使ってもらって構わないので、ぜひ、皆様のプログラミングの際に活用してください。

PSTR$()関数　数値を誤差ゼロで文字列に変換する

```
DEF PSTR$(A)
 VAR L,M,P,R$,S$,Z,SV=FALSE
 R$=STR$(A)
 IF CLASSIFY(A) || R$=="0" || R$=="-0" THEN RETURN R$
 S$=FORMAT$("%.340F",A)
 M=R$[0]=="-"
 Z=INSTR(S$,".")-M
 S$[0]=S$[0]*!M
 S$[Z]=""
 WHILE !VAL(S$[L])
  INC L
 WEND
 Z=Z-L-1
 IF Z<16 && Z>-6 THEN P=Z:Z=Z-P
 L=Z+341
 IF Z<0 THEN S$=RIGHT$(S$,L)
 P=(P>0)*P
 S$[P]=S$[P]+"."
 WHILE S$[L]=="0"
  DEC L
 WEND
 S$=LEFT$("-"*M+LEFT$(S$,L-(S$[L]==".")+1)+("e"+"+"*(Z>0)+ST
R$(Z))*!!Z+" "*SV*24,25)
RETURN S$
END
```

公開キー 【 NKADEE93 】

プチコン3号は通常は表示の際に小数第8位までで丸めたものが表示されるため正確な値を知りたければFOMAT$()で必要な桁数を指定して表示の必要があるし、絶対値の大きな数は0がたくさん表示されて非常に分かりづらいですが、このPSTR$()を使えば誰でも簡単に丸められていない分かりやすい表示が可能です。

PSTR$()は誤差ゼロで文字列に変換できるので文字列化したものをVAL()関数で数値化しても元の値から変化しないのでゲームのセーブデータを作る際に活用が可能です。(100万未満の整数以外はSTR$では誤差が発生する)

PSTR$()をセーブデータで活用したいという場合はリスト中のSV=FALSEをSV=TRUEに書き直せば25桁の文字列（25桁未満の場合は先頭にスペースが付く）で変換されます。これによって、複数の数値を文字列化した場合も先頭から25文字ずつ切り出していってVAL関数で数値化すれば誤差ゼロで文字列→数値の変換が可能です。

SUFFIX()関数　型を判別する

```
DEF SUFFIX(V)
 VAR A
 A=V>0<3
 IF A THEN V=0.5:A=V*2+1
 RETURN A
END
```

公開キー【 KAEEX3Z4 】　この公開キーはSUFFIXの他にERASE、INIT、INITA、INITSの合計5つの関数を含む

※この関数では整数型は1、実数型は2、文字列型は0を返す

これはプチコン3号では文字列と数値の比較演算は3を返すという裏ワザを私が発見したため実現可能になりました。この裏ワザを使わなければこのプログラムと比べてはるかに大規模なプログラムになります。

数値型か文字列型かさえ判別できれば良いのならばA=V>0<3だけでできます。（この場合は数値は1、文字列は0を返す）

配列全体の比較演算はできないためSUFFIX(A[0])のように引数に添え字付きで記述する必要があります。

SLOT()関数　動作しているスロットを取得する

```
DEF SLOT()
```

```
 VAR A$,B,I
 FOR I=0 TO 3
  A$=STR$(I)+":_SLOT"
  B=CHKVAR(A$)
  IF B THEN VAR(A$)=1
  _SLOT=0
  IF B && !VAR(A$) THEN BREAK
 NEXT
RETURN I
END
```

公開キー【 AEE3E8NV 】
※この関数を使用する場合はグローバル変数 _SLOT をあらかじめ用意しておく必要があります。

簡単に動作原理を説明しておくと指定したスロット番号のグローバル変数_SLOTに初期値1を入れてその後に_SLOTの値を0にした上で指定したスロット番号の_SLOTが0になっていればその指定したスロット番号が現在動作しているスロットとなるわけです。これは、動作しているスロットでは_SLOTの値は1から0に変わるけどそれ以外は変わらないためです。VAR(A$)=1という代入処理がないとたまたま他スロットに_SLOTというグローバル変数があり、さらに実体化した状態になっていてたまたま条件を満たしている値だった場合に誤判定をしてしまいます。

RVAL()関数　式を評価する

```
DEF RVAL(E$)
 PRGEDIT 1
 PRGDEL -1
 PRGSET "@R R="+E$+":RETURN
 USE 1
 GOSUB "1:@R"
RETURN VAR("1:R")
END
```

公開キー【 D4E1E73 】
※このRVAL()はワークエリアとしてスロット1を使用するためスロット1以外で実行してください。

プチコン3号には文字列を数値化するVAL()関数がありますがこれは定数にしか使用できません。例えばVAL("1+2")を実行しても数値化できず0を返すのですが、ちゃんと1+2を計算して3を返すのがこのRVAL()関数です。
プチコン3号で実行できるすべての演算子や関数（自作関数を含む）を計算した値を返すことができます。ただし、変数を使った式には対応していません。
なお、このRVAL()関数を使用すればINPUT A$:?RVAL(A$)というごく短いプログラムで電卓を作ることができます。プチコン3号はダイレクトモードを使えば簡単に電卓として計算が可能なのわざわざプログラムとして用意する必要はないですが、ある程度の拡張性を持たせて変数Xにも対応してダイレクトモードよりも使いやすくした「簡易電卓QSP」作ったのでRVAL()関数の使用例として参考にしてみてください。

SLEEP()関数　スリープ時間を取得

```
DEF SLEEPTIME(A)
VAR S,M=_SLEEPTIME
 A=A*1000
 _SLEEPTIME=MILLISEC
 S=_SLEEPTIME-M
 IF !!A*S>A THEN RETURN S RETURN 0
END
```

公開キー【 23NXDWQV 】
※この関数を使用する場合はグローバル変数 _SLEEPTIME をあらかじめ用意しておく必要があります。

プチコン3号においてMAINCNTはスリープ中は停止するのに対してMILLISECはスリープ中でも動作し続けます。これを利用することでスリープ時間（ミリ秒）が取得可能になります。引数は0の場合は初期化、0以外の時は最小検知スリープ時間（秒）となります。
使用するためにはまず最初にメインループ実行直前にSLEEPTIME(0)を実行して初期化しておきます。メイ

ンループ内で随時A=SLEEPTIME(1)を実行するとAにスリープ時間が入ります（スリープしてない時には0を返す)。通常は引数は1でも問題ないですが、ごく短い開閉時間を取得する場合は引数は0.1くらいの値を設定してください。

使用サンプルプログラム

```
VAR _SLEEPTIME
A=SLEEPTIME(0)
REPEAT
 A=SLEEPTIME(1)
 VSYNC
UNTIL A
?A/1000;"秒 スリープしたよ!"
```

　このサンプルプログラムを実行して本体を閉じて少し経って本体を開けばその間のスリープの時間を表示します。素早く開閉した際の時間を計測したければA=SLEEPTIME(1)の部分をA=SLEEPTIME(0.1)に変更すれば良いです。(私が試す限りどれだけ速く開閉してもスリープ時間が0.3秒を下回ることが無かったため)

2-5　ぜひ入手しておくべきプチコン3号プログラム

　プチコン3号では非常に多くのゲームやツールがユーザーの手によって作られていますがその中のごく一部をここで紹介します。
　任天堂のオンラインサービスが終了する2024年4月9日以降はダウンロードできなくなるし、それより前であってもバージョンアップによって公開キーが変更になってしまう可能性があるため興味がある作品はなるべく早めにダウンロードしておきましょう。これをご覧になった時点で、すでに公開キーが変更になっているもの公開を取り下げているものがある可能性がありますが、その点はご容赦ください。

（※作者名はすべて敬称略です）

ゲーム

アクションゲーム

芸夢狂人さんの「スネークワールド」をアレンジしたアクションゲーム		
CHIME and DASH!	ぷちぷち	8RNYW39J
おもらししないようにトイレに駆け込む横スクロールアクションゲーム		
ドウクツジマ	push	DK384E4S
探索型アクションゲーム。単体でeショップで販売もされているため面白さも折り紙付き		
走れアメジス	こりんご	Q3S3EJ9F
横スクロールアクションゲーム。スコアによって様々なご褒美CGが見れます。		
HAPPY X MAS	ぷちぷち	Q34E334D
サンタを操作してうまく家に着地してプレゼントを配ろう		
ぷちこちゃんX	まげ	MKK8VPZ6
オリジナルキャラのぷちこちゃんがやパンチやキックで障害物を払いのけるアクションゲーム		
雪のレインディア	すう	ZK343VNE
障害物を避けながら雪中でトナカイを走らせる疑似3Dゲーム		
リズミカルおもちつきKR MAX	やりのめ	55C3H3J4
太鼓の達人風の音ゲー。☆TAKA☆さんの「リズミカルおもちつき」の派生作品。		

シューティングゲーム

GIVERS-P3D	Bug太郎	735YW3L4
3Dポリゴンシューティング。ベーマガに掲載の「GIVERS」のリメイク		
光翼のリンドブルム	ToT k	CH3N3DH6
横スクロールシューティング。1ステージのみのループです。		
ストロベリー ストーム ストライカー	ゆのみ	NK5NY3KJ
2画面を使った本格的なシューティング		
PLANE2	SatoshiMcCloud	NKA8QPR1
ワイヤーフレーム描画による3Dシューティング		

レース・スポーツゲーム

F-ねこ	れい	KSNQEA1
F-ZERO風の疑似3Dレースゲーム（※エラーが出る場合はプログラム中のSQRをSQRTに置換してください）		
THE DRIVE	ぷちぷち	4KXKY3GF
プチコン3号の本格派疑似3Dレースゲーム。リアルな挙動を表現しています。		
VOLLEY BALL	りきすけ	QK483EJF
2人制のバレーボールゲームです。		

RPG

nRouge	neko800	QDSJ2X83
プチコン3号のローグ（不思議のダンジョン）のようなRPG		
まじっくすと～り～	るかか	RZVXH33J
キャラの掛け合いが楽しい本格派3DダンジョンRPG		

ツール・その他

お絵描き

色鉛筆	BlackSoft	55C3528E
色鉛筆風の線画に特化した軽量お絵かきソフト		
簡易毛筆	おちゃめ	9KS33X31
毛筆風お絵描きソフト。水墨画も描けます。超高速のフルカラーの混色表現が可能です。		
K2KPaint	くつきくれむ	4R83EXED
レイヤー対応。カケアミなどの漫画的な表現をするのに便利な機能もあり。		
juhsy	妄碌庵	2BC4738V
レイヤー対応。多彩な表現が可能（※高度サウンドユニットが必要。無しでも動く旧版もありDD53VKHE）		
Pics Limited	すう	1DVENVAE
本格的なお絵かきツール。水彩画のような表現も可能。		

音楽

NCF-MUSEQ	ナカノシマCF	C323R4
MML出力できるグリッド入力形式の音楽シーケンサ		
音楽つくろうぜ！	ナツキ	EKE3YXCX
MML作成支援ツール。五線に音符を並べて曲を作ることができる。		

通信

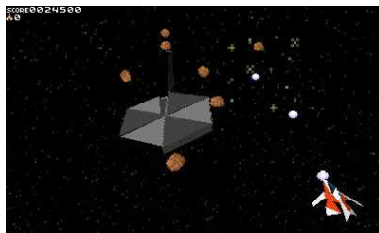
SENDFILE	あおごん	237YK334
プチコン3号のファイルをローカル通信で他の3DSに送る		
プチコン3号のプログラムをQRコード化してスマホやPCで読み取り可能にする		
PetitModem	れい	RSEE3D3
PCとプチコン3号の間で、 音波でファイルを転送するためのソフトウェア　http://rei.to/petitmodem.html		
ぷっちコンバーター3号	ゆうぱぱ	KANXDJM
画像を使ってプチコン3号からPCへデータを転送　https://tokui55.com/post-62		

それ以外

RMG IME	りゅうまご	MK334DAY
プチコン3号で漢字入力を可能にする日本語FEP		
e-Oscillo	すっとこ	7ECYA3DJ
マイク信号を入力とする使ってて「気持ち良い」オシロスコープ		
おちゃめQSPコレクション	おちゃめ	C3X3EJGF
QSPで作られたシンプルなゲームやツールが多数入ったお得なセット。（.MENUでメニューが立ち上がります）		
P3D-MODELER	Bug太郎	BBW3E88V
P3D-ENGINE用モデリングツール		

アルカナストーン

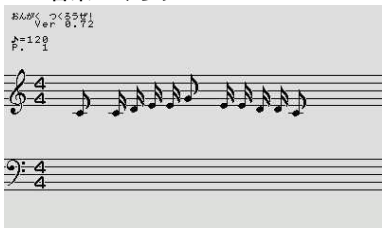
CHINE and DASH!!

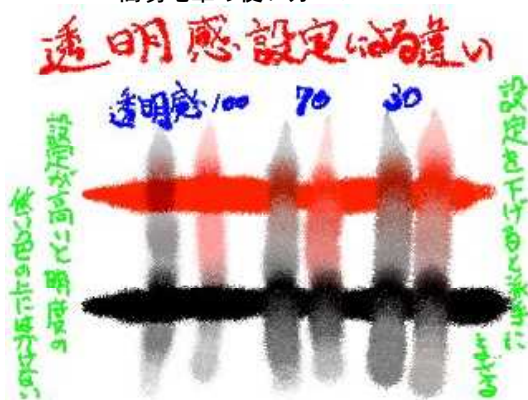
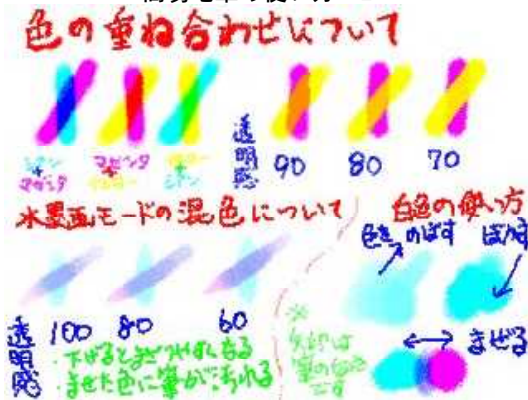
GIVERS-P3D

F-ねこ

nRouge

まじっくすと～り～

RMG IME

音楽つくろうぜ！

簡易毛筆

簡易毛筆の使い方　1

簡易毛筆の使い方　2

プチコン書き初め大会開催！！

2016年に第1回が行われたプチコン書き初め大会ですが、2024年の正月には第9回プチコン書き初め大会が開催されました。

第10回も2025年に開催予定です。上記の簡易毛筆の他、簡易毛筆シリーズで書かれた作品であればどれでも参加できます。詳細ルールは第9回の大会公式サイトを御覧になってください。

http://ochameclub.web.fc2.com/petitcom4/kakizome_9.htm

おまけ　プチコンBIG編

プチコンBIGはプチコン3号用に作られたプログラムの多くが動作するため**出先ではプチコン3号を使って自宅ではUSB**

キーボードや大画面モニタが使えるプチコンBIGでコーディングを行うという人も多いと思います。
　プチコン3号とプチコンBIGは公開キーも同じなのでプチコン3号専用、プチコンBIG専用の作品を除けば同じ公開キーでプチコン3号でもプチコンBIGでも動作可能です。プチコンBIGはメモリ8MBのプチコン3号と比べてメモリ128MBとメモリが非常に多く速度もNew3DSで動作するプチコン3号と比べても3～4倍程度高速であるため高速、大容量のプチコン3号として使えるだけではなくプチコンBIG用に拡張した機能を使うことでさらにすごい作品を作ることが可能です。

　プチコンBIGもプチコン3号と同じく4月9日以降はネットを使ったサービスが利用できなくなります。その場合はプチコン3号と同じく1画面プログラムを作ってスクリーンショットで公開するか音波を使ってPCへ転送するという方法があります。ただし、プチコン3号とプチコンBIGではマイクの仕様が異なるためプチコンBIG対応のプログラムを用意する必要があります。

　プチコンBIGはGRPの圧縮保存に対応しています。プチコン3号でも圧縮GRPの読み出しには対応しているのでプチコン3号で生成したGRPは一旦プチコンBIGで保存してやればGRPのサイズを数分の1～数10分の1に小さくできます。単色ラクガキの場合はプチコン3号で保存した場合はだと非圧縮で512KBのGRPファイルはプチコンBIGでは10KB以下になることがあります。ただし、この圧縮保存によってプチコン3号に手軽にファイル転送できるのはネットを使ったサービスが利用できる間に限ります。そのため、4月9日以降は使用できなくなるのでご注意ください。

最後に

　何度も書きますが2024年4月9日以降はネットを使った機能は使えなくなってしまいます。しかし、プチコンmkII、プチコン3号のソフトそのものは本体が壊れるまでは使えるのでぜひこれからも使い続けてみてください。終了後は確かに様々な制限が出てきますが、そのための対策も本書で書いているので問題ないと思います。

プチコンmkII、プチコン3号　Forever!!

　なお、本書に誤字脱字等のミスや補足などがある場合のために「プチコンmkII・プチコン3号を一生遊ぶ本　ver.1.1」サポートページを用意しています。ぜひ、御覧になってください。

http://ochameclub.web.fc2.com/CLUB/c101_1_1.htm

プチコンmkII・プチコン3号を
一生遊ぶ本　ver.1.1

ver.1.0　2022年12月31日発行
ver.1.1　2024年　3月　1日発行

制作・発行　　おちゃめくらぶ
発行人　　　　御茶目菜子（おちゃめ）

Website　　　おちゃめくらぶ
　　　　　　http://ochameclub.web.fc2.com/
X (twitter)　https://twitter.com/ochame___nako
pixiv　　　　https://pixiv.me/ochame_nako
e-mail　　　ochame_nako@hotmail.com

おちゃめくらぶ

2022,2024